

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

⑩ 日本国特許庁(JP)

⑪ 特許出願公開

⑫ 公開特許公報(A)

昭62-72059

⑬ Int. Cl.⁴

識別記号

庁内整理番号

⑭ 公開 昭和62年(1987)4月2日

G 06 F 15/20
3/02
3/023
15/203 0 1
3 7 0
3 1 0
3 0 1R-7010-5B
A-7218-5B
L-7218-5B
Y-7010-5B

審査請求 有 発明の数 1 (全8頁)

⑮ 発明の名称 混合オブジェクト文書の編集方法

⑯ 特 願 昭61-193030

⑰ 出 願 昭61(1986)8月20日

優先権主張 ⑱ 1985年9月23日 ⑲ 米国(US) ⑳ 779063

⑳ 発 明 者 バーバラ・エイ・パーカー
アメリカ合衆国テキサス州ラウンド・ロック、カーメル・ドライブ2800番地㉑ 発 明 者 アイリーン・エイチ・ヘルナンデス
アメリカ合衆国テキサス州オースチン、カウンスル・プラフ・ドライブ13304番地㉒ 発 明 者 ビバリー・エイチ・マチャート
アメリカ合衆国テキサス州オースチン、ドシルス・ドライブ11301番地㉓ 出 願 人 インターナショナル・ビジネス・マシーンズ・コーポレーション
アメリカ合衆国 10504 ニューヨーク州 アーモンク、(番地なし)

㉔ 代 理 人 弁理士 頓宮 孝一 外1名

明 細 書

1. 発明の名称 混合オブジェクト文書の編集方法

2. 特許請求の範囲

コンピュータ・ワークステーションで、混合したオブジェクト・タイプを含む文書を、インライン式で、かつ動的に編集するインテリジェント編集方法であって、

コンピュータ・ワークステーション上に、文書と、該文書の編集の際にユーザによって選択され得る一般的なアクションを少なくとも含むコマンド・バーを表示し、

ユーザによって、編集のために、該文書上のオブジェクトが選択されたか否かを判断し、

ユーザによって編集のために選択されたオブジェクトのタイプを判断し、

前記コマンド・バーに、ユーザによって編集のために選択されたオブジェクトのタイプに特有の編集アクションを表示する

ことを特徴とする混合オブジェクト文書の編集方法。

3. 発明の詳細な説明

A. 産業上の利用分野

本発明は、總括的には統合多重データの編集方法に関し、さらに詳しくは、混合したオブジェクト・タイプを含む文書を、インライン式で、かつ動的に編集するインテリジェント編集方法に関する。

B. 従来技術

今日のコンピュータ・プログラムの傾向として、様々なプログラムのアプリケーションの統合ということが挙げられる。統合がなければ、ユーザは、ワード・プロセッシング、データ・プロセッシング、およびグラフィック・アプリケーション用のアプリケーション・プログラムをばらばらに使用しなければならない。アプリケーション・プログラムがばらばらに使われた場合には、こうしたいくつかのアプリケーション・プログラムの出力を1つの文書に併合することが、しばしば困難になる。し

たがって、プログラム・アプリケーションを統合する目的は、コンピュータの簡単な操作でもって、所望の出力を得ることにあるといえる。

統合の最も簡単な形式は、同じファイルを共有し、同じまたは同様の機能を遂行するのに同じまたは同様のコマンドを用い、同じように動くようにデザインされた一連のアプリケーション・プログラムである。このような形式の統合は、比較的实现し易いけれども、そのファミリの個々のプログラムを同時に走行させることはできない。現在、最も一般的な統合ソフトウェアは、いわゆる多重機能(マルチプル・ファンクション)プログラムである。これは、様々なアプリケーションが1つのプログラムの中に組み込まれたものとして特徴づけられている。このようなプログラムによれば、一般的には、スクリーンはいくつかの異なるウィンドウに分割され、その各々があたかも別々のプログラムのように動くことができる。こうした多重機能プログラムは、代表的には、テキスト・アプリケーション、スプレッドシート・アプリケ

ーション、およびビジネス・グラフィック・アプリケーションを含む。すべてのアプリケーションが1つの共通のデータ・セットを共有するようなデータベース・アプリケーション環境に基づく統合技術は、多重機能プログラムと同様なものである。統合オペレーティング環境が適用されるときは、全く異なった方策が採られる。この方策では、個々のプログラムは情報を共有することができ、各自のウィンドウで、各々が同時にスクリーン上に出現できる。多重機能プログラムがパッケージの中にプログラムされたアプリケーションに限定されるのに対し、上記の方策では、アプリケーションを選択して組み合わせることができる。

IBM社では、個人用オンライン統合テキストエディタ(POLITE)を開発した。これは、複合文書のための簡単に使えるリアルタイム式のエディタ兼フォーマッタである。複合文書とは、イメージ、描画、グラフィック、図表、手書きのもの、テキスト、表、および数学的な記号を含むものである。エディタというのは、個々のアプリ

ケーションを呼び出す必要なしに、かつ、切取(カット)および貼り付け(ペースト)用のパツファを使うことなしに、統合された機能をインラインで処理できるべきであるというのがPOLITEの考え方である。Apple社のLISA(商標)コンピュータ、Macintosh(商標)コンピュータおよびXerox社のStar(商標)コンピュータは、いずれも複合文書を編集できる能力を持った統合システムである。しかしながら、これらのコンピュータでは共通して、まず、要求された機能の結果を切取用パツファに置き、それから、該結果を所望の場所で貼り付けるために文書エディタに戻すという方法が採られている。このやり方は、時間がかかり、しかも冗長である。

POLITEによれば、単一のアプリケーションの中から機能の統合がもたらされるけれども、編集機能は、明示的なユーザー・アクションによつて制御される。ユーザは、適用可能なアクションが選択される前に、オブジェクトに適用されるメニューを表示するか、または、オブジェクト選

択後に、コマンド・ライン上にアクションをタイプするかしなければならない。

特開昭61-62170号公報に開示されている発明は、POLITEシステムに基づくアプリケーション複合エディタの改良を志向している。アプリケーション複合エディタは、POLITEと同様、テキストだけでなく、イメージ、グラフィック、表、注釈および手書きのものを含む複合文書用の、簡単に使えるリアルタイム式のエディタ・フォーマッタである。このアプリケーション複合エディタは、動的に編集可能な形で、単一ページ上ですべてのデータを互いに関連させて統合することができる。すべてのデータ・タイプを同一文書内で生成することができ、テキストはグラフィックおよび表のまわりに配置される。エディタにおけるすべてのデータはページに存在し、すべてのページは複合文書内に存在する。エディタは、データ・オブジェクトがページに存在し、データはそのデータ・オブジェクトに存在するというページ・レイアウトの考え方で動く。すべての

ページは文書オブジェクト内に存在する。データ・オブジェクトによつては、その中に付加的なオブジェクトを持つことがある。オブジェクトは、そのページでユーザが操作することのできるデータ特有のエンティティである。他のオブジェクト的に存在するオブジェクトは、定義されたオブジェクト・セットの境界内に存在する。すべてのオブジェクトは明示的、すなわち認識可能で選択可能なエンティティである。すべてのオブジェクトは、そのページの指定された境界内に存在する。この境界は、1つのオブジェクト・セットの境界として定義される。例えば、テキスト文字は、ライン・オブジェクト・セットまたはパラグラフ・オブジェクト・セットの境界内に存在し、矩形は、グラフィック・オブジェクトの境界内に存在し、セルは表オブジェクト・セットの境界内に存在する。

C. 発明が解決しようとする問題点

本発明とPOLITEにおけるエディタとの主な相違点は、編集環境を制御するのはエディタであつて、ユーザではないということである。エディ

タは、選択されたオブジェクト・タイプと、オブジェクトおよびアクションの選択に用いられるカーソルの現在の位置の、一方または両方に基づいて、表示するコマンドを決定する。ユーザが、明示的にコマンドの表示を要求することはない。エディタは、各文書ページを、テキスト、グラフィック、イメージまたは表データを含むことができ、そして、該ページ、または該文書の他のページ、または該文書の外の他のどのオブジェクト・セットとも関連を持つことのできるオブジェクト・セットによつて占められたエリアとして管理する。

本発明は、明示的なユーザ・コマンドよりも、オブジェクトのタイプおよび環境についての内的な知識に基づいて、オブジェクトおよび属性（アトリビュート）に対するアクションを実行する混合オブジェクトの編集方法の提供を目的とする。

本発明の他の目的は、オブジェクト、アトリビュートおよびコマンド実行の操作を、ユーザにとって明快なものとする簡単なユーザ・インタフェースを提供することにある。

本発明のさらに他の目的は、ばらばらのウィンドウを使つたり、入力ラインを屈曲したりすることなしに、ページ上でインライン式に、すべてのオブジェクトを編集する方法を提供することにある。

本発明のさらに他の目的は、文書内の混合オブジェクトの一般的な編集方法であつて、別のアプリケーションをロードしたり、明示的なコマンドを指定したりする必要のないものを提供することにある。

D. 問題点を解決するための手段

本発明によれば、上述のアプリケーション概念エディタは、ユーザとの対話、および多重データ／アクション特有サブルーチンを扱うコントローラを含む。コントローラは、前記サブルーチンを通して、オブジェクトの操作を管理し、コマンドの実行の経路指定を行う。コントローラは、ユーザ入力をモニタし、すべてのメニューとメッセージを表示するとともに、文書のフォーマツティング（形式化）、オブジェクトのポジショニング

（位置づけ）、簡単なテキスト編集、MOVE、COPY、DELETE、SEARCH/REPLACE、GOTO、GET、CREATEおよび状況、スケール、コマンド・ラインの表示等の一般的な機能をすべて扱う。オブジェクトの編集方法は、選択されたオブジェクトのタイプにかかわらず、同じである。ユーザはオブジェクトを選択し、該オブジェクトに適用されるアクションを指定する。例えば、パラグラフ・オブジェクトを選択することにより、テキストがパラグラフに挿入される。次に、テキストのカーソル位置から打鍵を行う。ヘッダとフッタは似た方法で更新される。すなわち、ヘッダまたはフッタのオブジェクトが選択され、カーソル位置から変更が指示される。多くのワード・プロセッサで要求されるように、そのような変更を別の編集ウィンドウで指定する必要はなく、POLITEで要求されるように、コマンド・ライン上で指定する必要もない。

エディタは、どのオブジェクト・タイプでも有効である一般的なコマンドのセットを含むメニュー

一を提供する。これらのコマンドに含まれるのは、GET、UNDO/REDO、CREATE、DESCRIBE、SEARCH、GOTO、DELETE、MOVEおよびCOPYである。エディタは、オブジェクトが選択されたらいつでも、表示中の一般コマンドに加えて、オブジェクトに特有のコマンドを表示する。例えば、グラフィック・オブジェクトが選択された場合、一般コマンドと同時にROTATE、SHAPEおよびSMOOTHが表示される。しかし、表オブジェクトが選択されたならば、ROTATE、SHAPE、SMOOTHはCLEAR、VIEW、SORTに置き換えられる。オブジェクトの選択が引き金となつて、エディタは、表示されるべきコマンドのセットを自動的に変更する。ユーザは、エディタに対して、あるコマンド・セットから次のコマンド・セットに変更するよう、明示的に命令してやる必要はない。

コントローラは、指定されたオブジェクトとアクションを使つて、どんな処理が必要かを判断し、

の例としてパラグラフ3とライン4、描画グラフィック・オブジェクトの例として矩形5と手書き線図6、ビジネス・グラフィック・オブジェクトの例として円グラフ7、そして、表オブジェクトの例としてスプレッド・シート8が表示されている。コマンド・バー9は、ページ2のどのオブジェクトにも有効な一般コマンドを示している。ここで、ユーザは、コマンド・バー9上の任意のコマンド、またはページ2のオブジェクトを表わすアイコン10を含む、ページ2上の任意のオブジェクトを選択することができる。

第2A、2B、2C、2Dおよび2E図は、エディタによつてサポートされるオブジェクト・タイプのそれぞれに有効なコマンドを示している。第2A図のコマンド・バー11は、すべての文書オブジェクトに共通するコマンドを含む。第2B図のコマンド・バー12は、一般コマンドに加えて、テキスト・オブジェクトに特有のコマンドであるMERGE、SPLIT、およびCONVERTを含む。第2C図のコマンド・バー13には、

必要に応じて適当なサブルーチンを自動的に呼び出す。このような操作は、ユーザにとって全く明快なものである。このようなデザインによれば、ユーザはもはや異なるオブジェクト・タイプやページ上のデータの併合に要求される処理について悩まなくてもよくなる。フレキシビリティを損なうことなく、表はグラフィックの隣に置けるし、グラフィックはテキストの隣に置けるのである。
E. 実施例

本発明をより詳しく理解するため、まず、図面の第1図を参照する。この図には文書ページ2が表示されており、すべての点がアドレス指定可能（オール・ポイント・アドレスابل、APA）、すなわちピント・マツプされる表示装置の総集ウインドウ1に表示中であると仮定する。なお、表示装置の制御は、スタンド・アローンのコンピュータまたはホストにつながれたコンピュータによつて行われる。ユーザがページ2を含む文書の総集を要求すると、該ページは総集ウインドウ1に現れる。ページ2には、テキスト・オブジェクト

一般コマンドとともに、描画グラフィック・オブジェクトに特有のコマンドであるROTATE、SHAPEおよびSMOOTEが表示されている。第2D図のコマンド・バー14には、一般コマンドに加えて、ビジネス・グラフィック・オブジェクトに特有のコマンドであるROTATE、SHAPE、およびCONVERTが表示されている。第2E図のコマンド・バー15には、表オブジェクトに特有のコマンドであるCLEAR、VIEW、およびSORTが含まれている。総集のためにオブジェクト・タイプが選択されたときはいつでも、システムは、そのオブジェクトに特有のコマンド・バーの1つを自動的に表示する。例えば、第1図の矩形5が選択されると、システムはコマンド・バー13を表示するし、パラグラフ3またはライン4が選択されると、システムはコマンド・バー12を表示する。システムは、選択されたオブジェクトのタイプと、該オブジェクトの総集用に表示する必要のあるコマンドとを常に知っている。

次に、第3図のフローチャートを参照する。これには、オブジェクト混合文書の編集のプロセスが説明されている。まず、オペレータが文書の編集を指示する。すると、システムは、該文書の編集に有効なコマンドとともに、該文書が既に存在している場合はその第1ページを、該文書がこれから作成される場合には空白のページを、表示する。ここで、オペレータは、編集ウィンドウ内の任意の場所にポインタを位置させ、コマンドまたはオブジェクトの一方を選択する。このようなアクションは、ブロック21、22に記されている。オブジェクトが選択されたならば、制御の流れはブロック23を経てブロック24へ至る。編集コマンドが選択されたならば、制御はブロック31へ渡る。ブロック24にて、システムは、オブジェクトの選択に先立つて編集アクションが選択されていたか否かをチェックする。アクションが選択済ならば、システムは、ブロック27でアクションを処理する。編集アクションの処理後、ブロック28にて、システムは、文書ページの再フォーマットの必要の有無をチェックする。ページの再フォーマッティングと該ページのオブジェクトのレイアウトは、ブロック30で起る。ブロック29では、文書は編集ウィンドウに再表示され、制御はブロック22に戻る。

ブロック24において、編集アクションが選択されていなかった場合、制御はブロック25に渡る。そこで、システムは、表示中のコマンドが選択されたオブジェクトに対して有効か否かをチェックする。コマンドが有効でない場合、システムは選択済のオブジェクトに特有のアクションを含むコマンド・バーを自動的に表示する。有効なコマンド・バーを表示した後、制御はブロック22へ戻る。

編集アクションが選択された場合、システムは、ブロック31にて、アクションが文書編集を終了させるのか否かをチェックする。アクションが終了アクションでないならば、制御はブロック32に移る。ブロック32にて、システムは、該アクションの選択に先立つてオブジェクトの選択が行

われたか否かをチェックする。オブジェクトの選択が行われていた場合、システムは、ブロック27でアクションを処理し、ブロック28で再フォーマットの必要性をテストし、ブロック30で再フォーマット、ブロック29で文書の再表示を行う。次に、制御はブロック22へ戻る。オブジェクトの選択が行われていなかった場合も、制御はブロック22へ戻り、オペレータの選択を待つことになる。

以下で説明するのは、知的かつ動的な、混合データ・タイプを含む文書のエディタ兼フォーマットを実現する。ワークステーションまたはホストのプロセッサによって使用可能なアプリケーション・プログラムについてである。このプログラム(第1表参照)は、ソースおよびマシン・コードを記述することのできるプログラム・デザイン言語(PDL)で記されている。以下の説明では、オペレータが、システムに対して、顕著な特性を持つ編集オブジェクトのセット、および該オブジェクトと特性に適用可能なアクションのセットを含

む、文書編集環境の提供を要求したものと仮定する。

第1表

(○文書は、オペレータによって定義される初期編集状態に従って表示されている。○)

```

REPEAT
  CALL GET_PTR_POS(I_LOC, I_SEL)
  IF AN EDIT OBJECT HAS BEEN SELECTED
    THEN IF AN EDIT ACTION HAS PREVIOUSLY BEEN
      SELECTED
    THEN CALL EXECUTECHD(CMD_TYPE, OBJ_TYPE)
    IF REFORMATTING OF PAGE REQUIRED
      THEN CALL FORMAT_PAGE
    ENDIF
    CALL REDISPLAY
  ELSE CALL CHECK_CHDS(OBJ_TYPE, CHDS_OKAY)
    IF NOT CHDS_OKAY
      THEN CALL DISPLAY_CHDS(OBJ_TYPE)
    ENDIF
  ENDIF

```

```

ELSE (*EDIT ACTION SELECTED*)
  IF EDIT ACTION <> TERMINATE
    THEN IF AN EDIT OBJECT HAS PREVIOUSLY BEEN
      SELECTED
    THEN CALL EXECUTECHD(CMD_TYPE, OBJ_TYPE)
    IF REFORMATTING OF PAGE REQUIRED
      THEN CALL FORHAT_PAGE
    ENDIF
    CALL REDISPLAY
  ENDIF
ENDIF
UNTILL CMD_TYPE = TERMINATE

```

オペレータが編集セッションの開始を指示した後、類似される文書は、初期編集状態の条件に従って表示される。文書の第1ページが、表示スクリーン上の編集ウィンドウの中に位置される。CREATE、DELETE、MOVE、COPY等の一般コマンドからなるメニューも表示される。ここで、オペレータは、ページ・ヘッダ、ページ・

TYPE、OBJ__TYPE))。コマンド・タイプ(CMD__TYPE)とオブジェクト・タイプ(OBJ__TYPE)は、該ルーチンに対して、コマンドの実行を完了するための、付加的なオブジェクトに特有のルーチン呼び出す必要の有無の判断を用いることのできる情報を与える。該コマンドの処理が終わった後、エディタは、再フォーマットの必要のあるページの有無を判断し、ページをフォーマットするルーチンを自動的に呼び出す(CALL FORMAT__PAGE)。次に、文書ウィンドウの内容を更新して表示するルーチンが呼び出される(CALL REDISPLAY)。FORMAT__PAGEルーチンは、オブジェクト・タイプ、フォーマット特性および既存の関係すべてに従って、ページ上の各オブジェクトを位置づける。コマンドの選択が以前に行われていなかった場合は、表示中のコマンドが選択されたオブジェクト・タイプ(OBJ__TYPE)にとって適切であることを確かめるルーチンが呼び出される(CALL CHECK__

フッタ、テキスト・パラグラフ、グラフィック、ピクチャのような文書内の任意のオブジェクト、または任意のコマンドを選択することができる。

次に、システムは、オペレータの入力のモニタを開始する。ポインタの位置を定めるとともに、オブジェクトまたはコマンドが選択されたか否かを判断するルーチンが呼び出される(CALL GET__PTR__POS(I__LOC, I__SEL))。パラメータI__LOCは、ポインティング装置のX、Y位置を与え、パラメータI__SELは、オブジェクトまたはコマンドが選択されたか否かを指定する。エディタは、カーソル位置をウィンドウ内のエリアと関連づける。ウィンドウは、編集オブジェクトと関連づけられるものである。システムは、入力がコマンドでない場合は、メニュー入力をオブジェクトとして扱う。

オブジェクトが選択され、かつコマンドが以前に選択されていた場合には、オブジェクトに対するアクションを処理するルーチンが呼び出される(CALL EXECUTECHD(CMD__T

CMDS(OBJ__TYPE, CMDS__OKAY))。フラグCMDS__OKAYは、選択されたオブジェクトタイプ用の有効なコマンドを決定し、該コマンドを編集ウィンドウに表示するルーチン呼び出す必要の有無を判断するのに用いられる(CALL DISPLAY__CMDS(OBJ__TYPE))。システムは、コマンドまたは他のオブジェクトが選択されるまで、オペレータの入力をモニタし続ける。

非終了コマンドが選択され、かつ以前にオブジェクトが選択されていた場合には、EXECUTECHDルーチンが呼び出され、オブジェクトに対するアクションを処理する。必要に応じて、FORMAT__PAGEとREDISPLAYが自動的に呼び出される。オブジェクトの選択が以前に行われていなかった場合は、処理がさらに要求されるまで、システムは、オペレータの入力をモニタし続ける。QUITまたはENDのような終了コマンドが選択されると、編集セッションは終わる。

ROTATE	SHAPE	CONVERT	GET	UNDO/REDO	PRINT	PASTE	END
SHOW	CREATE	DESCRIBE	SEARCH	GOTO	DELETE	MOVE	COPY

ビジネス・グラフィック・コマンド
第2D図

CLEAR	VIEW	SORT	GET	UNDO/REDO	PRINT	PASTE	END
SHOW	CREATE	DESCRIBE	SEARCH	GOTO	DELETE	MOVE	COPY

表コマンド
第2E図

